From Systems to Ecosystems: Rethinking Adaptive Safety

David Halasz
Masaryk University
Brno, Czech Republic
halasz@mail.muni.cz

ABSTRACT

The evolution of software systems into more complex ecosystems creates new challenges in ensuring their safe and secure behavior. As the complexity of software ecosystems is inherently higher than regular systems, existing safety mechanisms are no longer reliable in their context. This paper introduces a research path towards adaptive safety mechanisms that can support the degree of dynamism and high level of uncertainty introduced by these systems of systems. Our planned approach is to use runtime trust evaluation as a decision factor when enabling or disabling safety features on demand.

KEYWORDS

autonomous ecosystems, adaptive safety, software architecture, trust, security

ACM Reference Format:
David Halasz. 2022. From Systems to Ecosystems: Rethinking Adaptive Safety. In *17th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS '22), May 18–23, 2022, PITTSBURGH, PA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3524844.3528067

1 INTRODUCTION

There is an ongoing evolution of software systems forming more complex systems of systems, called ecosystems [33]. This evolution can also be perceived across other domains, including autonomous cyber-physical systems where it is bringing a higher degree of autonomy and self-adaptation [8]. Multiple systems working together as a whole inherently increases the probability of encountering unpredictable situations, and reacting to uncertainties and dynamic changes becomes complicated as well. In this context, existing quality mechanisms in these software ecosystem architectures are no longer capable of dealing with the dynamicity of context changes and autonomous behavior needs [8].

This makes the evolution in self-adaptive mechanisms for such systems inevitable, putting an increasing focus on the trust among the ecosystem components, as well as the need of safety-assurance mechanisms to ensure safety under the risk of an untrusted component entering the ecosystem. The term safety here is understood as defined by McKinley et al. [30] as "the ability of a distributed application and its parts to continue operating in a safe manner during and after a transformation". Furthermore, as defined by Banerjee et al. [4] as "avoidance of hazards to the physical environment". The term trust has many definitions depending on the domain of science in which it is being used [10]. In our case the definitions borrowed from Philosophy [24], Psychology [34] and Organizational Management [29] are the most compatible with our focus on autonomous systems.

According to Liu et al. [27], reputation-based trust can be an effective countermeasure for securing machine-to-machine communications, arguing that this can be taken further to secure any kind of interaction between machines. Cioroaica et al. [14] proposes that the reputation of a system or system component can be based on observation of its runtime behavior and its compliance to its Digital Twin, verified with runtime trust-assessment techniques. That is a vision that we are complementing with the work proposed here.

The goal of this paper is to present our vision towards a novel safety-assurance approach for dynamic autonomous ecosystems under trust considerations. Section 2 summarizes the problem for the reader, pointing out the key challenges in the domain. Related work is presented and evaluated in the Section 3 and Section 4 extends this by introducing our proposed solution. The plan for evaluation and validation of this new idea is drafted in the Section 5 and our expected contributions are listed in the Section 6. Finally, Section 7 describes the current status of our research.

2 PROBLEM STATEMENT

Due to the unprecedented complexity of autonomous ecosystems, mechanisms ensuring their safe behavior cannot be simply borrowed from the domain of autonomous systems. Individual components can dynamically join or leave the ecosystem at any time and this behavior is unpredictable. The key challenge is to adapt the corresponding runtime safety mechanisms to the inherent uncertainty about the assessed level of trust towards a component, i.e. how much a component can be trusted.

Safety assurance in dynamic autonomous ecosystems faces numerous challenges. In the context of our work, they include:

- **Intentional vs unintentional behavior:** When enforcing safety, it is necessary to distinguish between intentional and unintentional behavior. A system might be malfunctioning or relying on incorrect information (data from sensors, communication lag, external factors, etc.) and thus negatively affecting its peers or the environment. In a more harmful case, the autonomous system can be intentionally designed to maximize the inflicted damage on a certain target or disrupt the ecosystem as a whole by its actions or by propagating false information. It is not clear which mitigation technique should be selected in a situation where the intent of the system is uncertain. Informing a system with malicious intent about its behavior can invest this system with tactical advantage in achieving its goals. Meanwhile, enforcing safety mechanisms

SEAMS '22, May 18–23, 2022, PITTSBURGH, PA, USA

ACM ISBN 978-1-4503-9305-8/22/05…$15.00
https://doi.org/10.1145/3524844.3528067

on a system that could correct its behavior in the knowledge of the right piece of information can cause a stall.

- **False positives and false negatives:** When trust in a system is assumed incorrectly, both scenarios carry their own danger. If a system is marked as trusted, but it should not be it is a *false-positive* case in the context of trust and it can exploit this to maximize damage during a hostile maneuver. A *false-negative* case is when a system is incorrectly marked as untrusted and it is being restricted from accessing features of the ecosystem. Reacting to these scenarios by enabling or disabling safety features can lead to either unnecessarily limited systems or non-mitigated safety incidents.
- **Supervision awareness:** A system can be designed to adapt its behavior when it detects that it is being probed in a confined environment. If the data earned under supervision are further used to determine its level of trust, the given system can detect that it is no longer monitored and start causing damage. Even the usage of a mitigation technique or a safety mechanism could give out hints to the system about being supervised. With this in mind, techniques ensuring the safety of the ecosystem should be designed in a way to not give any hint about any form of confinement or supervision. However, in some cases when the malicious behavior of a system has been qualified as unquestionably unintentional, informing the system of this fact can actually flip it back into its normal and safe operation.
- **Feedback loops:** If a system by mistake classifies another system as malicious and triggers its safety mechanism, the other system can consider this as a hostile operation and also enable some features that ensure safety. In response, the first system would increase the reach of its safety mechanism and the two peers would get into a feedback loop that can get them into a stall. When considering more than two systems, this can lead to a stall on the level of the whole ecosystem. Moreover, malicious systems can be intentionally designed to disrupt the ecosystem this way. Safety features should be designed to be aware of these potential situations and should provide a way out.

The aim of our research is to propose a novel safety-assurance approach for dynamic autonomous ecosystems that, compared to existing approaches, addresses all these challenges. This research is being proposed within the context of a research group investigating trust in dynamic autonomous ecosystems. We believe that combining trust with novel dynamic adaptive safety mechanisms that ensure safety in uncertain situations is a hard research problem worth pursuing.

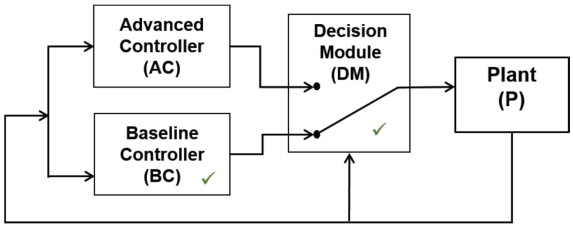
3 RELATED WORK

Extensive body of research exists towards ensuring safety and security in the specific domains of autonomous systems, especially autonomous vehicles. According to Jahan et al. [20] there are substantial research gaps in existing solutions, including:

- *"Uncertainties in modeling"* – that can be linked to the four challenges emphasized above.
- *"Accomplishing higher goals through cooperation and collaboration"* – that can be addressed with the perspective of the overall dynamic autonomous ecosystems, i.e. a set of systems working together while achieving safety and security on both the individual and the collective level.

Simplex architecture. Simplex is a software architecture for real-time safety-cricical and high-assurance complex systems [37, 38]. Its core idea is to use a pair of controllers, one that is advanced and complex and the one baseline that is simple but reliable. Normally the advanced one is enabled and a decision module is constantly monitoring its behavior. In case of a failure, the decision module gives the control to the baseline controller until the problem is resolved. It is possible to use this architecture to compose complex autonomous systems with multiple simplexes interconnected [32, 43].

Figure 1: The simplex architecture [32]



While the philosophy of "using simplicity to control complexity" is an interesting approach to achieve runtime safety, it might not produce good results in uncertain situations. For example in case of a false-positive scenario, the limitations inflicted on a system could be unnecessarily severe. Moreover, the use of certified baseline controllers could be prone to the supervision awareness described in the previous chapter.

Isolated environments. In other domains, virtualization [36], sandboxing [19] and similar strategies isolating untrusted code from the production environment have been used. This offered inspiration [3] for autonomous systems as well. However, they can not be applied directly as there are major differences in these domains. A software sandbox, for example, would prevent a cyber-physical system from causing damage in the cyber part of the ecosystem, however, it would be less successful in mitigating the attacks targeting the physical world. In the context of autonomous ecosystems, the applicability becomes even more challenging as the member systems not necessarily have the means to enforce other systems into a software sandbox.

Autonomous vehicles. The main focus of research in the last couple of years in the area of autonomous systems is indisputably around autonomous vehicles. In this domain, in the context of safety, the research is mainly focused on communication security, countermeasures upon failure or following an attack [7, 15], vehicle platooning [2] and various forms of collision avoidance [26]. While research about in vehicle platooning is in the domain of autonomous ecosystems and they deal with safety of the whole ecosystem, they are not applicable to other types and use cases of software ecosystems. In the case of an autonomous vehicle that has been intentionally designed to cause damage by either sending false sensory data or causing collision with another vehicle, these countermeasures might not always be applicable. Mitigation strategies that handle attacks

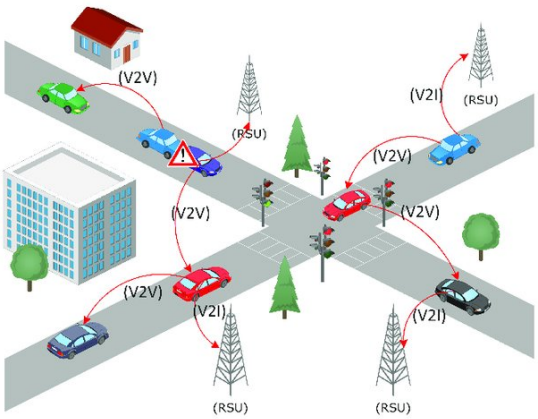
on sensors or collision avoidance might help in some cases. However, they were not designed for these kinds of scenarios. Furthermore, mechanisms that would focus on ensuring safety of the autonomous ecosystem as a whole seem to be not available yet.

Wireless networks. Research in this area is rather focused on security than on safety. This is due to the fact that the main purpose of any computer network is communication and in this context security [35, 40] is the dominating concern. Even though research around these communication security issues do not provide a solution to the problem stated in the previous chapter, some of the techniques from the domain can be still of use.

A similar architecture to our envisioned dynamic software ecosystems are a subset of wireless networks. A self-organizing wireless mesh network [18, 21] as a whole can be interpreted as a system of systems working together to achieve a common goal. Some mitigation techniques that target malicious nodes on a network [16, 31, 41] can be partially applied in our broader domain of autonomous ecosystems.

Another interesting field is the security in wireless sensor networks. In addition to information about the network, the data acquired from sensors [9] can be also leveraged. This extra information can be used to better coordinate mitigation strategies. However, due to the lack of means to manipulate the physical world, the safety features can only have a reach in the cyber part of an ecosystem.

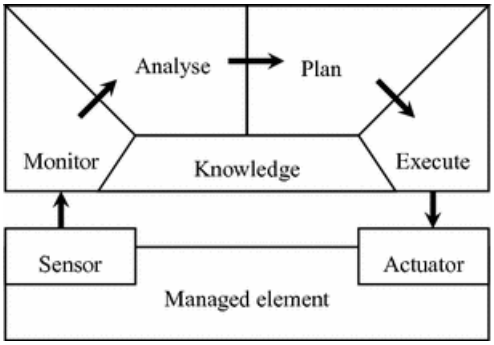
Figure 2: Vehicular Area Network [39]



Finally, Vehicular Area Networks (VANETs) are an intersection of wireless ad-hoc networks and autonomous vehicles. They are a combination of Road Side Units (RSU) placed on fixed locations alongside the road with autonomous vehicles. However, there are approaches to render RSUs unnecessary by moving their functionality into the vehicles [42]. While the research on VANET security [17] is also noteworthy, the main focus of our interest is in the safety features they can provide on the ecosystem level [28]. Unfortunately, as these networks do not have means to enforce the safe behavior in the physical world, their safety mechanisms are only applicable partially in our context.

MAPE-K feedback loops. The usage of *Monitor - Analyze - Plan - Execute* based on *Knowledge* feedback loop is a proven engineering approach to implement self-adaptation [23, 25]. As it can also be applied as a continuous loop, it is possible to cover the whole lifecycle of an autonomous system that is capable of handling uncertain situations. There are approaches to specify these feedback loops as in abstract stateful languages, that allow validation and verification of various adaptation scenarios [1].

Figure 3: The MAPE-K feedback loop



While MAPE-K loops are a great tool to describe and model solutions for uncertain scenarios, we believe, they are hard to design for complex systems. This becomes even more challenging in the context of autonomous ecosystems, where an unknown number of member systems can join or leave at any time.

Runtime models. The model-driven engineering (MDE) methodology emphasises the creation and usage of models in software development [22]. The *models@run.time* approach inspires from this methodology and extends it by leveraging such models at runtime [6]. This makes a software system more self-aware, equipping it with additional capabilities for decision-making and self-adaptiveness based on circumstances that were not anticipated during design-time. Research has been done in the context of using these runtime models to decide whether an adaptation in an autonomous system is necessary or not [5].

Due to the dinamicity of an autonomous ecosystem, it might be not possible to construct its model. In case this model would exist, it is still unclear where should it be stored and which member systems could access it. A system with malicious intent could use this model to its own advantage and try to exploit its design to disrupt the ecosystem.

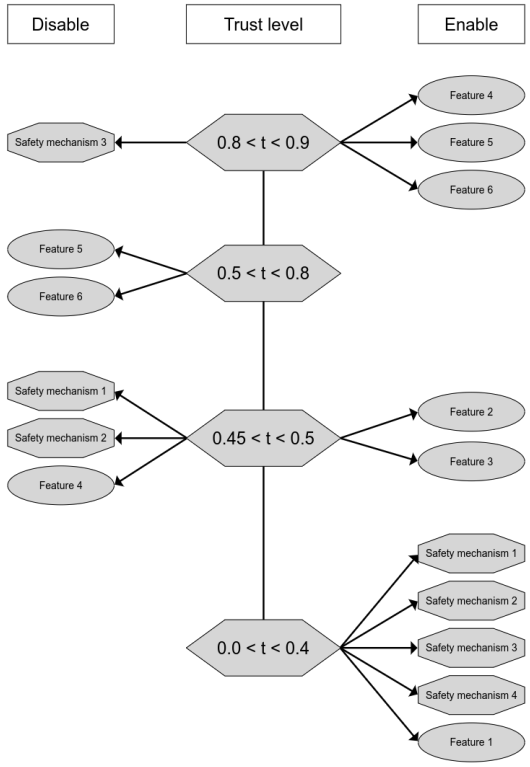
4 PROPOSED SOLUTION

The complexity and dynamicity of autonomous software ecosystems makes ensuring their safe and secure behavior a challenging task. The core of the problem is to design an architecture that, despite the described uncertainties and challenges, is able to maintain safety both on the individual component level and on the level of the ecosystem as a whole.

Our view is that runtime evaluated dynamic trust can be used as a decision factor when enforcing safety in a complex autonomous ecosystem. When a new system joins the ecosystem, it is handled by its peers as a black box. Each of these systems would assess a level of trust towards this new system individually, based on their interaction with the given system. However, they can also rely on trust levels propagated by other systems, based on how much they trust these other systems.

Any proposed trust model has to be constructed in a way that it provides a highly granular output. Having a simple binary output from a method would only allow a simple binary triggering of any safety feature. In case of an error in the trust evaluation (false positive or false negative) the mitigation would have a negative effect on the ecosystem from the aspect of its safety as a whole. On the other hand, additional granularity reduces the negative impact of a failed assessment of trust.

Figure 4: Example scenario: actions to take based on trust



Based on the level of trust, a system can conceal or expose its vulnerable features and enable or disable its safety mechanisms. Due to the granularity of the trust value, any of these reactions enforcing safety can be done selectively and gradually. For example if the level of trust is rising, the set of available features can be also expanded with it while disabling certain safety mechanisms as well. As an example, figure 4 presents a decision graph for when to enable or disable various features and safety mechanisms based on the assessed trust level. This architecture can reduce the inherent risk from uncertainties and also successfully tackle the following scenarios:

- If a system with malicious intent reaches a high trust, there are still safety features enabled and in the moment the trust drops due to suspicious behavior, the gradual enablement of the safety features can quickly mitigate it.
- In case the behavior of a system is falsely classified as suspicious and the safety features are gradually enabled, it still has enough room for a limited operation.
- When the malicious behavior is unintentional, the mitigation would only affect this specific behavior and allow the rest of the system to function normally.
- In case of a feedback loop between two systems, the trust propagating from other systems should be capable of breaking this loop as it provides fresh and correct information to both systems.

5 PLAN FOR EVALUATION AND VALIDATION

The validation of our work will be conducted by the experimental comparison with the state-of-the-art approaches. We envision that a simulation of an autonomous ecosystem might be the right approach, where we can integrate our technique and compare how it performs against existing solutions. Moreover, we would like to reach out to automotive companies and try coordinating our efforts with at least one of them to validate our proposed solution on realistic case studies. The idea should be applicable to any autonomous ecosystem, however, our primary focus is to ensure the safety of a larger-scale transportation infrastructure, such as a whole city.

6 EXPECTED CONTRIBUTIONS

We expect to contribute to the state of the art with:

- A method for adaptive safety that addresses the challenges stated above.
- A software architecture focusing on this method, capable of ensuring safety in autonomous ecosystems with adaptive mechanisms.
- A software framework that implements this architecture and can be applied in complex ecosystems.

7 CURRENT STATUS

The focus of our research team is around trust, especially in the domains of Autonomous Ecosystems and Software Engineering. The team already has some promising results on trust building between autonomous vehicles via Digital Twins [11–14]. The research in the domain of adaptive safety is still in its very early phase and not yet published.

We believe that this idea would change the way we perceive safety and evolve into a set of tools for promoting safe autonomous ecosystems.

ACKNOWLEDGMENTS

The work has been prepared under the supervision and guidance of Barbora Buhnova.

REFERENCES

[1] Paolo Arcaini, Elvinia Riccobene, and Patrizia Scandurra. 2015. Modeling and Analyzing MAPE-K Feedback Loops for Self-Adaptation. In *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. 13–23. https://doi.org/10.1109/SEAMS.2015.10

[2] Jakob Axelsson. 2016. Safety in Vehicle Platooning: A Systematic Literature Review. *IEEE Transactions on Intelligent Transportation Systems* 18 (08 2016), 1–13. https://doi.org/10.1109/TITS.2016.2598873

[3] Stanley Bak, Karthik Manamcheri, Sayan Mitra, and Marco Caccamo. 2011. Sandboxing Controllers for Cyber-Physical Systems. In *2011 IEEE/ACM Second International Conference on Cyber-Physical Systems*. 3–12. https://doi.org/10.1109/ICCPS.2011.25

[4] Ayan Banerjee, Krishna K. Venkatasubramanian, Tridib Mukherjee, and Sandeep Kumar S. Gupta. 2012. Ensuring Safety, Security, and Sustainability

of Mission-Critical Cyber–Physical Systems. *Proc. IEEE* 100, 1 (2012), 283–299. https://doi.org/10.1109/JPROC.2011.2165689

[5] Matthias Barkowsky, Thomas Brand, and Holger Giese. 2021. Improving Adaptive Monitoring with Incremental Runtime Model Queries. In *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. 71–77. https://doi.org/10.1109/SEAMS51251.2021.00019

[6] Nelly Bencomo, Robert France, Betty H.C. Cheng, and Uwe Aßmann (Eds.). 2014. *Models@run.time: foundations, applications, and roadmaps.* Springer, Germany. https://doi.org/10.1007/978-3-319-08915-7 Dagstuhl Seminar 11481 on models@run.time held in November/December 2011.

[7] Siham Bouchelaghem, Abdelmadjid Bouabdallah, and Mawloud Omar. 2020. Autonomous Vehicle Security: Literature Review of Real Attack Experiments. In *The 15th International Conference on Risks and Security of Internet and Systems*. Paris, France. https://hal.archives-ouvertes.fr/hal-03034640

[8] Rafael Capilla, Emilia Cioroaica, Barbora Buhnova, and Jan Bosch. 2021. On Autonomous Dynamic Software Ecosystems. *IEEE Transactions on Engineering Management* (2021), 1–15. https://doi.org/10.1109/TEM.2021.3116873

[9] Xiangqian Chen, Kia Makki, Kang Yen, and Niki Pissinou. 2009. Sensor network security: a survey. *IEEE Communications Surveys Tutorials* 11, 2 (2009), 52–73. https://doi.org/10.1109/SURV.2009.090205

[10] Jin-Hee Cho, Kevin Chan, and Sibel Adali. 2015. A Survey on Trust Modeling. *ACM Comput. Surv.* 48, 2, Article 28 (oct 2015), 40 pages. https://doi.org/10.1145/2815595

[11] Emilia Cioroaica, Barbora Buhnova, Thomas Kuhn, and Daniel Schneider. 2020. Building trust in the untrustable. In *2020 IEEE/ACM 42nd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE, 21–24.

[12] Emilia Cioroaica, Stanislav Chren, Barbora Buhnova, Thomas Kuhn, and Dimitar Dimitrov. 2019. Towards creation of a reference architecture for trust-based digital ecosystems. In *Proceedings of the 13th European Conference on Software Architecture-Volume 2*. 273–276.

[13] Emilia Cioroaica, Stanislav Chren, Barbora Buhnova, Thomas Kuhn, and Dimitar Dimitrov. 2020. Reference Architecture for Trust-Based Digital Ecosystems. In *2020 IEEE International Conference on Software Architecture Companion (ICSA-C)*. IEEE, 266–273.

[14] Emilia Cioroaica, Thomas Kuhn, and Barbora Buhnova. 2019. (Do not) trust in ecosystems. In *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*. IEEE, 9–12.

[15] Jin Cui, Lin Shen Liew, Giedre Sabaliauskaite, and Fengjun Zhou. 2019. A review on safety failures, security attacks, and available countermeasures for autonomous vehicles. *Ad Hoc Networks* 90 (2019), 101823. https://doi.org/10.1016/j.adhoc.2018.12.006 Recent advances on security and privacy in Intelligent Transportation Systems.

[16] S. Desilva and R.V. Boppana. 2005. Mitigating malicious control packet floods in ad hoc networks. In *IEEE Wireless Communications and Networking Conference, 2005*, Vol. 4. 2112–2117 Vol. 4. https://doi.org/10.1109/WCNC.2005.1424844

[17] Richard Gilles Engoulou, Martine Bellaïche, Samuel Pierre, and Alejandro Quintero. 2014. VANET security surveys. *Computer Communications* 44 (2014), 1–13. https://doi.org/10.1016/j.comcom.2014.02.020

[18] Albrecht J. Fehske, Ingo Viering, Jens Voigt, Cinzia Sartori, Simone Redana, and Gerhard P. Fettweis. 2014. Small-Cell Self-Organizing Wireless Networks. *Proc. IEEE* 102, 3 (2014), 334–350. https://doi.org/10.1109/JPROC.2014.2301595

[19] Chris Greamo and Anup Ghosh. 2011. Sandboxing and Virtualization: Modern Tools for Combating Malware. *IEEE Security Privacy* 9, 2 (2011), 79–82. https://doi.org/10.1109/MSP.2011.36

[20] Farha Jahan, Weiqing Sun, Quamar Niyaz, and Mansoor Alam. 2019. Security Modeling of Autonomous Systems: A Survey. *ACM Comput. Surv.* 52, 5, Article 91 (sep 2019), 34 pages. https://doi.org/10.1145/3337791

[21] Ryszard Katulski, Jacek Stefański, Jaroslaw Sadowski, Slawomir Ambroziak, and Bożena Miszewska. 2009. *Self-Organizing Wireless Monitoring System for Containers*. 164–172. https://doi.org/10.1007/978-3-642-03841-9_15

[22] Stuart Kent. 2002. Model Driven Engineering. In *Integrated Formal Methods*, Michael Butler, Luigia Petre, and Kaisa Sere (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 286–298.

[23] J.O. Kephart and D.M. Chess. 2003. The vision of autonomic computing. *Computer* 36, 1 (2003), 41–50. https://doi.org/10.1109/MC.2003.1160055

[24] Bernd Lahno. 1999. *Ethical Theory and Moral Practice* 2, 4 (1999), 433–435. http://www.jstor.org/stable/27504108

[25] Philippe Lalanda, Julie A McCann, and Ada Diaconescu. 2013. *Autonomic computing: principles, design and implementation.* Springer Science & Business Media.

[26] Guofa Li, Yifan Yang, Tingru Zhang, Xingda Qu, Dongpu Cao, Bo Cheng, and Keqiang Li. 2021. Risk assessment based collision avoidance decision-making for autonomous vehicles in multi-scenarios. *Transportation Research Part C: Emerging Technologies* 122 (2021), 102820. https://doi.org/10.1016/j.trc.2020.102820

[27] Ling Liu, Margaret Loper, Yusuf Ozkaya, Abdurrahman Yasar, and Emre Yigitoglu. 2016. Machine to Machine Trust in the IoT Era. In *Proceedings of the 18th International Conference on Trust in Agent Societies - Volume 1578* (Singapore, Singapore) *(TRUST'16)*. CEUR-WS.org, Aachen, DEU, 18–29.

[28] Xiaomin Ma, Jinsong Zhang, Xiaoyan Yin, and Kishor S. Trivedi. 2012. Design and Analysis of a Robust Broadcast Scheme for VANET Safety-Related Services. *IEEE Transactions on Vehicular Technology* 61, 1 (2012), 46–61. https://doi.org/10.1109/TVT.2011.2177675

[29] Roger C. Mayer, James H. Davis, and F. David Schoorman. 1995. An Integrative Model of Organizational Trust. *The Academy of Management Review* 20, 3 (1995), 709–734. http://www.jstor.org/stable/258792

[30] P.K. McKinley, S.M. Sadjadi, E.P. Kasten, and B.H.C. Cheng. 2004. Composing adaptive software. *Computer* 37, 7 (2004), 56–64. https://doi.org/10.1109/MC.2004.48

[31] Ambidi Naveena and Katta Rama Linga Reddy. 2018. Malicious node prevention and mitigation in MANETs using a hybrid security model. *Information Security Journal: A Global Perspective* 27, 2 (2018), 92–101. https://doi.org/10.1080/19393555.2017.1415399 arXiv:https://doi.org/10.1080/19393555.2017.1415399

[32] Dung Phan, Junxing Yang, Matthew Clark, Radu Grosu, John Schierman, Scott Smolka, and Scott Stoller. 2017. A Component-Based Simplex Architecture for High-Assurance Cyber-Physical Systems. In *2017 17th International Conference on Application of Concurrency to System Design (ACSD)*. 49–58. https://doi.org/10.1109/ACSD.2017.23

[33] Pilar Rodríguez, Alireza Haghighatkhah, Lucy Ellen Lwakatare, Susanna Teppola, Tanja Suomalainen, Juho Eskeli, Teemu Karvonen, Pasi Kuvaja, June M. Verner, and Markku Oivo. 2017. Continuous deployment of software intensive products and services: A systematic mapping study. *Journal of Systems and Software* 123 (2017), 263–291. https://doi.org/10.1016/j.jss.2015.12.015

[34] Julian B. Rotter. 1980. Interpersonal trust, trustworthiness, and gullibility.

[35] Linda S. Rutledge and Lance J. Hoffman. 1986. A survey of issues in computer network security. *Computers & Security* 5, 4 (1986), 296–308. https://doi.org/10.1016/0167-4048(86)90050-7

[36] Jyotiprakash Sahoo, Subasish Mohapatra, and Radha Lath. 2010. Virtualization: A Survey on Concepts, Taxonomy and Associated Security Issues. In *2010 Second International Conference on Computer and Network Technology*. 222–226. https://doi.org/10.1109/ICCNT.2010.49

[37] D. Seto, B. Krogh, L. Sha, and A. Chutinan. 1998. The Simplex architecture for safe online control system upgrades. In *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No.98CH36207)*, Vol. 6. 3504–3508 vol.6. https://doi.org/10.1109/ACC.1998.703255

[38] Lui Sha. 2001. Using simplicity to control complexity. *IEEE Software* 18, 4 (2001), 20–28. https://doi.org/10.1109/MS.2001.936213

[39] Syed Sarmad Shah, Asad Malik, Anis Ur Rahman, Sohail Iqbal, and Samee Khan. 2019. Time Barrier-Based Emergency Message Dissemination in Vehicular Ad-hoc Networks. *IEEE Access* PP (01 2019), 1–1. https://doi.org/10.1109/ACCESS.2019.2895114

[40] Muhammad Shoaib Siddiqui. 2007. Security Issues in Wireless Mesh Networks. In *2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07)*. 717–722. https://doi.org/10.1109/MUE.2007.187

[41] Ming-Yang Su, Kun-Lin Chiang, and Wei-Cheng Liao. 2010. Mitigation of Black-Hole Nodes in Mobile Ad Hoc Networks. In *International Symposium on Parallel and Distributed Processing with Applications*. 162–167. https://doi.org/10.1109/ISPA.2010.74

[42] Ozan K. Tonguz and Wantanee Viriyasitavat. 2013. Cars as roadside units: a self-organizing network solution. *IEEE Communications Magazine* 51, 12 (2013), 112–120. https://doi.org/10.1109/MCOM.2013.6685766

[43] Prasanth Vivekanandan, Gonzalo Garcia, Heechul Yun, and Shawn Keshmiri. 2016. A Simplex Architecture for Intelligent and Safe Unmanned Aerial Vehicles. In *2016 IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. 69–75. https://doi.org/10.1109/RTCSA.2016.17